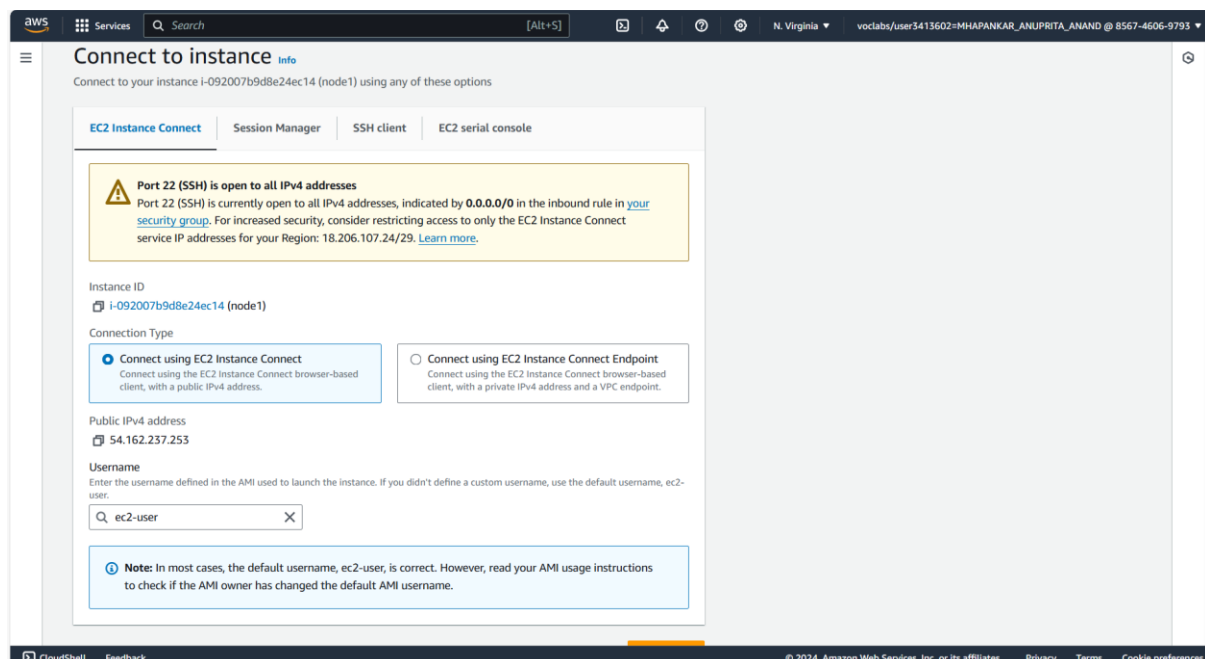
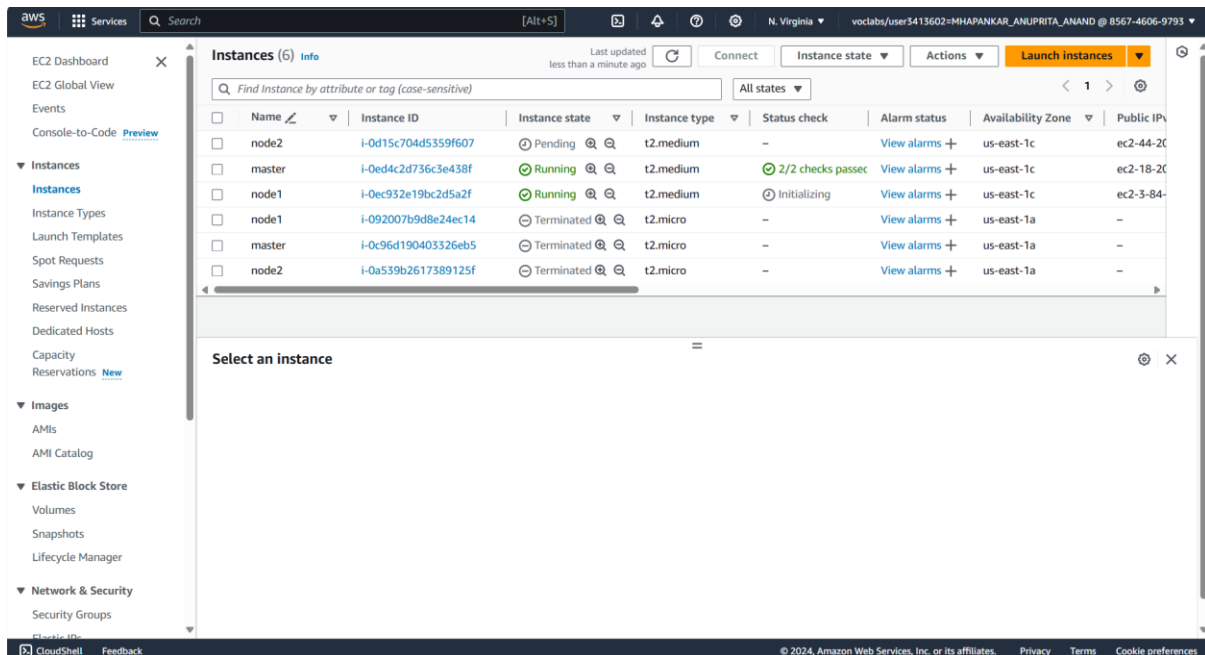


## Experiment 4

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Step 1: Go to AWS Academia in services select EC2 and create 3 instance with instance type t2.medium and names as node1, node2 and master



Step 2: Create a new key pair and name it as myKey1 and download as .pem file. Then, open command prompt and go to the directory where the key is downloaded and run the following command

```
chmod 400 myKey1.pem
```

ssh -i myKey1.pem [ec2-user@3.88.13.120](mailto:ec2-user@3.88.13.120)

Repeat the steps for node1, master and node2

```
ec2-user@ip-172-31-30-94:~$ ssh -i myKey1.pem ec2-user@3.88.13.120
Warning: Permanently added '3.88.13.120' (ED25519) to the list of known hosts.
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Wed Sep 18 17:26:06 2024 from 18.206.107.29
[ec2-user@ip-172-31-30-94 ~]$
```

Step 3: Select and connect each instance and run the following commands inside the console of each instance.

sudo su

yum install docker -y

systemctl start docker

docker --version

yum repolist

Amazon Linux 2023  
https://aws.amazon.com/linux/amazon-linux-2023

```
(ec2-user@ip-172-31-33-243 ~)$ sudo su
[root@ip-172-31-33-243 ec2-user]# yum install docker -y
Last metadata expiration check: 0:10:44 ago on Wed Sep 18 13:13:43 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
<b>Installing:</b>				
docker	x86_64	25.0.6-1.amzn2023.0.2	amazonlinux	44 M
<b>Installing dependencies:</b>				
containerd	x86_64	1.7.20-1.amzn2023.0.1	amazonlinux	35 M
iptables-libs	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux	401 k
iptables-nft	x86_64	1.8.8-3.amzn2023.0.2	amazonlinux	183 k
libcgroup	x86_64	3.0-1.amzn2023.0.1	amazonlinux	75 k
libnetfilter_conntrack	x86_64	1.0.8-2.amzn2023.0.2	amazonlinux	58 k
libnftnl	x86_64	1.0.1-19.amzn2023.0.2	amazonlinux	30 k
libnftnl	x86_64	1.2.2-2.amzn2023.0.2	amazonlinux	84 k
pkgconf	x86_64	2.5-1.amzn2023.0.3	amazonlinux	83 k
runC	x86_64	1.1.13-1.amzn2023.0.1	amazonlinux	3.2 M

Transaction Summary  
Install 10 Packages

i-0a539b2617389125f (node2)  
PublicIPs: 107.21.35.198 PrivateIPs: 172.31.33.243

```
aws Services Search [Alt+S] N. Virginia voclabs/user3413602:MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 4/10
Installing : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 5/10
Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Installing : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 7/10
Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Installing : libcgrouper-3.0-1.amzn2023.0.1.x86_64 9/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.


Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgrouper-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64 10/10

Installed:
containerd-1.7.20-1.amzn2023.0.1.x86_64      docker-25.0.6-1.amzn2023.0.2.x86_64      iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgrouper-3.0-1.amzn2023.0.1.x86_64    libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnftnl-1.0.1-19.amzn2023.0.2.x86_64      libnftnl-1.2.2-2.amzn2023.0.2.x86_64    pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-33-243 ec2-user]# systemctl start docker
[root@ip-172-31-33-243 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bdc
[root@ip-172-31-33-243 ec2-user]#
```

i-Oa539b2617389125f (node2)  
PublicIPs: 107.21.35.198 PrivateIPs: 172.31.33.243

Step 4: Now, go to the following link <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/> and scroll down and select Red-Hat based distributions tab copy all the commands on by one in each console of instance.

 **kubernetes**

Documentation Kubernetes Blog Training Partners Community Case Studies Versions English Search this site

Search this site

Documentation

Getting started

Learning environment

Production environment

Container Runtimes

Installing Kubernetes with deployment tools

Bootstrapping clusters with kubeadm

Installing kubeadm

Troubleshooting kubeadm

Creating a cluster with kubeadm

Customizing components with the kubeadm API

Options for Highly Available Topology

Debian-based distributions

Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's"/SELINUX=enforcing"/SELINUX=permissive/' /etc/selinux/config
```

Caution:

- Setting SELinux in permissive mode by running `setenforce 0` and `sed ...` effectively disables it. This is required to allow containers to access the host filesystem; for example, some cluster network plugins require that. You have to do this until SELinux support is improved in the kubelet.
- You can leave SELinux enabled if you know how to configure it but it may require settings that are not supported by kubeadm.

2. Add the Kubernetes `yum` repository. The `exclude` parameter in the repository definition ensures that the packages related to Kubernetes are not upgraded upon running `yum update` as there's a special procedure that must be followed for upgrading Kubernetes. Please note that this repository have packages only for Kubernetes 1.31; for other Kubernetes minor versions, you need to change the Kubernetes minor version in the URL to match your desired minor version (you should also check that you are reading the documentation for the version of Kubernetes that you plan to install).

Edit this page

Create child page

Create documentation issue

Print entire section

Before you begin

Verify the MAC address and product\_uuid are unique for every node

Check network adapters

Check required ports

Swap configuration

Installing a container runtime

Installing kubeadm, kubelet and kubect

Configuring a cgroup driver

Troubleshooting

What's next

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

[root@ip-172-31-33-243 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                           Amazon Linux 2023 repository
kernel-livepatch                      Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-33-243 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-33-243 ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core/stable/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core/stable/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[root@ip-172-31-33-243 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes
Dependencies resolved.
54 kB/s | 9.4 kB | 00:00

Package Architecture Version Repository Size
-----
Installing:
kubeadm x86_64 1.31.1-150500.1.1 kubernetes 11 M
kubectl x86_64 1.31.1-150500.1.1 kubernetes 11 M
kubectl x86_64 1.31.1-150500.1.1 kubernetes 15 M
Installing dependencies:
conntrack-tools x86_64 1.4.6-2.amzn2023.0.2 amazonlinux 208 k

i-0a539b2617389125f (node2)
PublicIPs: 107.21.35.198 PrivateIPs: 172.31.33.243

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

Transaction test succeeded.
Running transaction
Preparing : kubernetes-cni-1.5.1-150500.1.1.x86_64 1/1
Installing : cri-tools-1.31.1-150500.1.1.x86_64 1/9
Installing : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 2/9
Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Installing : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 4/9
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 5/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 6/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 7/9
Installing : kubectl-1.31.1-150500.1.1.x86_64 8/9
Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64 9/9
Verifying : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 9/9
Verifying : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 1/9
Verifying : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 2/9
Verifying : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 3/9
Verifying : cri-tools-1.31.1-150500.1.1.x86_64 4/9
Verifying : kubeadm-1.31.1-150500.1.1.x86_64 5/9
Verifying : kubectl-1.31.1-150500.1.1.x86_64 6/9
Verifying : kubelet-1.31.1-150500.1.1.x86_64 7/9
Verifying : kubernetes-cni-1.5.1-150500.1.1.x86_64 8/9
Verifying : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64 cri-tools-1.31.1-150500.1.1.x86_64 kubeadm-1.31.1-150500.1.1.x86_64
kubectl-1.31.1-150500.1.1.x86_64 kubelet-1.31.1-150500.1.1.x86_64 kubernetes-cni-1.5.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-33-243 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service - /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-33-243 ec2-user]#

i-0a539b2617389125f (node2)
PublicIPs: 107.21.35.198 PrivateIPs: 172.31.33.243

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Step 5: Now, run the following command in the mater instance -  
kubeadm init

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602:MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

[root@ip-172-31-93-102 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0918 14:21:55.805697 28020 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by
kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder: "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-93-102.ec2.internal kubernet.es.default kubernet.es.default.svc kubernet.es.default.svc.c
luster.local] and IPs [10.96.0.1 172.31.93.102]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-93-102.ec2.internal localhost] and IPs [172.31.93.102 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-93-102.ec2.internal localhost] and IPs [172.31.93.102 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"

i-Oed4c2d736c3e438f (master)
PublicIPs: 18.208.183.159 PrivateIPs: 172.31.93.102

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Step 6: Now, run the following commands in master instance's console –

- `mkdir -p $HOME/.kube`  
`sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`  
`sudo chown $(id -u):$(id -g) $HOME/.kube/config`
- `export KUBECONFIG=/etc/kubernetes/admin.conf`
- `kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \`  
`--discovery-token-ca-cert-hash`  
`sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818`

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602:MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818

[root@ip-172-31-93-102 ec2-user]# mkdir -p $HOME/.kube
[root@ip-172-31-93-102 ec2-user]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[root@ip-172-31-93-102 ec2-user]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-93-102 ec2-user]# export KUBECONFIG=/etc/kubernetes/admin.conf
[root@ip-172-31-93-102 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
[preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
[ERROR FileAvailable--etc-kubernetes-kubelet.conf]: /etc/kubernetes/kubelet.conf already exists
[ERROR Port-10250]: Port 10250 is in use
[ERROR FileAvailable--etc-kubernetes-pki-ca.crt]: /etc/kubernetes/pki/ca.crt already exists
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
to see the stack trace of this error execute with '--v=5 or higher
[root@ip-172-31-93-102 ec2-user]#

i-Oed4c2d736c3e438f (master)
PublicIPs: 18.208.183.159 PrivateIPs: 172.31.93.102

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Step 7: Run this command in node1 and node2 -

`kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \`

--discovery-token-ca-cert-hash  
sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

Installing : contrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Running scriptlet: contrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 8/9
Installing : kubect1-1.31.1-150500.1.1.x86_64 9/9
Running scriptlet: kubect1-1.31.1-150500.1.1.x86_64 9/9
Verifying : contrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
Verifying : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
Verifying : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Verifying : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 4/9
Verifying : cri-tools-1.31.1-150500.1.1.x86_64 5/9
Verifying : kubeadm-1.31.1-150500.1.1.x86_64 6/9
Verifying : kubect1-1.31.1-150500.1.1.x86_64 7/9
Verifying : kubelet-1.31.1-150500.1.1.x86_64 8/9
Verifying : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Installed:
contrack-tools-1.4.6-2.amzn2023.0.2.x86_64      cri-tools-1.31.1-150500.1.1.x86_64      kubeadm-1.31.1-150500.1.1.x86_64
kubect1-1.31.1-150500.1.1.x86_64              kubelet-1.31.1-150500.1.1.x86_64      kubernetes-cni-1.5.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

complete!
[root@ip-172-31-95-221 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service -> /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-95-221 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vg5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
[preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info ConfigMap: Get "https://172.31.93.102:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
to see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-95-221 ec2-user]#

i-Od15c704d5359f607 (node2)
PublicIPs: 44.201.192.9 PrivateIPs: 172.31.95.221
```

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

Installing : contrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Running scriptlet: contrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64 7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64 7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64 8/9
Installing : kubect1-1.31.1-150500.1.1.x86_64 9/9
Running scriptlet: kubect1-1.31.1-150500.1.1.x86_64 9/9
Verifying : contrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
Verifying : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
Verifying : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
Verifying : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64 4/9
Verifying : cri-tools-1.31.1-150500.1.1.x86_64 5/9
Verifying : kubeadm-1.31.1-150500.1.1.x86_64 6/9
Verifying : kubect1-1.31.1-150500.1.1.x86_64 7/9
Verifying : kubelet-1.31.1-150500.1.1.x86_64 8/9
Verifying : kubernetes-cni-1.5.1-150500.1.1.x86_64 9/9

Installed:
contrack-tools-1.4.6-2.amzn2023.0.2.x86_64      cri-tools-1.31.1-150500.1.1.x86_64      kubeadm-1.31.1-150500.1.1.x86_64
kubect1-1.31.1-150500.1.1.x86_64              kubelet-1.31.1-150500.1.1.x86_64      kubernetes-cni-1.5.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

complete!
[root@ip-172-31-94-95 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service -> /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-94-95 ec2-user]# kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vg5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818
[preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info ConfigMap: Get "https://172.31.93.102:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
to see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-94-95 ec2-user]#

i-Oec932e19bc2d5a2f (node1)
PublicIPs: 3.84.157.220 PrivateIPs: 172.31.94.95
```

Step 8: Run the following command in master instance console -  
kubect1 get nodes



## Step 11: The website is live

